

# Reporte Técnico

Prototipo de sistema de identificación de estudiantes basado en reconocimiento facial dentro de la Escuela Superior de Cómputo para evitar la suplantación de identidad durante la aplicación de Exámenes a Título de Suficiencia mediante el uso de credenciales y dispositivos móviles

Trabajo terminal A-138

---

De la Cruz De la Cruz Alejandra  
Flores Esquivel Luis Antonio  
Huertas Ramírez Daniel Martin  
Jiménez Rodríguez José Alfredo

Escuela Superior de Cómputo, IPN

17 de noviembre de 2024





<b>1. Problemática</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>3</b>
2.1. Reconocimiento facial . . . . .	3
2.2. Redes neuronales . . . . .	3
2.3. Evaluaciones a Título de Suficiencia (ETS) . . . . .	3
2.3.1. Procedimiento para realizar un ETS . . . . .	4
2.4. Códigos QR . . . . .	4
2.4.1. Anatomía de un código QR . . . . .	4
2.4.2. Fiabilidad de los códigos QR . . . . .	7
2.5. Aplicación móvil . . . . .	8
2.5.1. Aplicaciones nativas . . . . .	8
2.5.2. Aplicaciones Web . . . . .	9
2.5.3. Aplicaciones Híbridas . . . . .	10
2.5.4. Aplicaciones Progresivas Web Apps (PWA) . . . . .	10
2.6. Sistema operativo . . . . .	11
2.7. Lenguaje de programación . . . . .	11
2.8. Framework . . . . .	11
2.8.1. Jetpack compose . . . . .	11
2.8.2. XML . . . . .	12



---

## Índice de figuras

---

2.1. Patrones de detección de posición. . . . .	4
2.2. Patrones de alineación. . . . .	5
2.3. Patrones de temporización. . . . .	5
2.4. Información sobre la versión. . . . .	6
2.5. Información del formato. . . . .	6
2.6. Código de corrección de datos y errores. . . . .	7
2.7. Márgenes. . . . .	7
2.8. Tipo de aplicaciones móviles. . . . .	8



---

## Índice de cuadros

---

2.1. Comparación tipos de aplicaciones móviles. . . . .	11
---------------------------------------------------------	----





# CAPÍTULO 1

## Problemática

*Presentar la problemática*



Este capítulo está enfocado en detallar la información esencial para el entendimiento del Trabajo Terminal, además de explicar y establecer las tecnologías que se usarán para el desarrollo de este. Para comenzar

#### 2.1. Reconocimiento facial

#### 2.2. Redes neuronales

#### 2.3. Evaluaciones a Título de Suficiencia (ETS)

En el Instituto Politécnico Nacional (IPN), incluyendo unidades académicas como la Escuela Superior de Cómputo (ESCOM), la acreditación de cada unidad de aprendizaje se realiza semestralmente a través de 3 evaluaciones ordinarias. Si un alumno no acredita alguna unidad de aprendizaje, tendrá la oportunidad de presentar una evaluación extraordinaria. Estos procedimientos están detallados en el programa de estudios y se encuentran especificados en el calendario académico [?].

El alumno que no logre acreditar una o más de las unidades de aprendizaje en la que se haya inscrito podrá optar por acreditarlas mediante Evaluación a Título de Suficiencia, de acuerdo con lo establecido en el Artículo 39 del Reglamento Interno del IPN, que señala:

“La evaluación del aprendizaje se llevará a cabo a través de exámenes ordinarios, extraordinarios y a título de suficiencia, cuyos requisitos y procedimientos de elaboración, presentación y exención, así como de otros mecanismos de evaluación continua, se realizarán en los términos que fijen los planes y programas de estudio, el presente Reglamento y los reglamentos respectivos.”

Existen dos rondas de ETS:

- ETS Ordinario: Esta es la primera oportunidad que tiene el alumno para acreditar la materia en la que no obtuvo una calificación aprobatoria. Los ETS ordinarios generalmente se aplican al finalizar el semestre, permitiendo al estudiante demostrar sus conocimientos sin necesidad de repetir el curso completo.

- ETS Especiales: Si el alumno no acredita la materia en el ETS ordinario, puede optar por presentar un ETS Especial. Esta es la segunda oportunidad que tiene el alumno para pasar la materia. Esta evaluación adicional suele programarse el primer viernes del nuevo semestre, brindando al estudiante una opción rápida para regularizar sus situación académica y continuar avanzando en su plan de estudios.

### 2.3.1. Procedimiento para realizar un ETS

- Pagar en caja, verificar que estén correctos los siguientes datos: Nombre, Boleta, Carrera y Número de unidades de aprendizaje.
- Acudir a ventanilla de gestión escolar para generar créditos en el “SAES”.
- Una vez generados los créditos, inscribe las unidades de aprendizaje en la página del “SAES”.
- Entregar en ventanilla de gestión escolar, el comprobante de inscripción de ES generador por SAES, y el recibo de pago para dar fin a la inscripción al ETS.
- Acudir el día y la hora establecida en el calendario [?].

## 2.4. Códigos QR

Un código QR es un tipo de código de barras bidimensionales que solo se puede leer con teléfonos inteligentes u otros dispositivos dedicados a la lectura de estos códigos. Cuando se lee un código QR, los dispositivos se conectan directamente a mensajes de texto, correos electrónicos, sitios web, números de teléfono, etc [?].

### 2.4.1. Anatomía de un código QR

Patrones de detección de posición



Figura 2.1: Patrones de detección de posición.

Los patrones de detección de posición se encuentran en las tres esquinas del código. Gracias a ellos, el escáner puede reconocer y leer el código QR rápidamente. Estos marcadores indican la dirección en la que se imprimió el código QR y ayudan a su identificación y orientación.

Patrones de alineación



Figura 2.2: Patrones de alineación.

Usados para corregir la distorsión del código QR en superficies curvas. El tamaño y la cantidad de los patrones de alineación pueden variar según el volumen de la información almacenada en el código.

Patrones de temporización



Figura 2.3: Patrones de temporización.

La alternancia de los módulos negros y blancos del código QR determina el sistema de información, también llamado cuadrícula de datos. Con estas líneas, el escáner reconoce la matriz de datos.

## Información sobre la versión



Figura 2.4: Información sobre la versión.

Estos marcadores indican cuál de las 40 versiones del código QR está siendo usada. Normalmente las versiones utilizadas son de 1 a 7.

## Información del formato



Figura 2.5: Información del formato.

Contiene información sobre la tolerancia a los errores y el patrón del enmascaramiento de datos. La información sobre el formato facilita el escaneo del código.

### Código de corrección de datos y errores



Figura 2.6: Código de corrección de datos y errores.

El sistema de corrección de errores del código QR almacena toda la información y comparte el espacio con los módulos de corrección de errores, que permiten reconstruir los datos perdidos.

### Márgenes



Figura 2.7: Márgenes.

Los márgenes, o también llamado zona quieta, alrededor del código QR son similares al espacio blanco en un diseño, proporcionan estructura y una mejor comprensión. Pero, ¿cómo? Para que el software de escaneo identifique bien el límite del código QR de sus alrededores, los márgenes son vitales.

## 2.4.2. Fiabilidad de los códigos QR

Los códigos QR están diseñados para mantener la información legible, aunque estén oscuros o dañados. Esto se logra mediante la compensación de errores, es decir, insertando la información varias veces. Con un alto nivel de seguridad, los códigos QR pueden leerse incluso si un tercio de su información es ilegible. Esto hace que los códigos QR sean muy fiables a la hora de guardar información.

Para la elaboración de nuestro TT, el código QR desempeña un papel fundamental en la corroboración de la identidad de los estudiantes. Las credenciales escolares al contener un código QR se puede aprovechar la capacidad para

almacenar y transmitir información de manera segura y rápida. Este enfoque nos permite verificar la información del estudiante mediante un simple escaneo, reduciendo el tiempo necesario para corroborar la identidad. Al escanear el código QR en la credencial. El sistema accede a los datos del alumno, lo cual permite verificar si coincide con la persona que se presenta al ETS.

## 2.5. Aplicación móvil

Una aplicación móvil, es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente. A diferencia de las aplicaciones diseñadas para computadoras de escritorio, las aplicaciones móviles se alejan de los sistemas de software integrados.

Debido a los recursos de hardware limitados de los primeros dispositivos móviles, las aplicaciones móviles evitaban la multifuncionalidad. Sin embargo, incluso si los dispositivos que se utilizan hoy en día son mucho más sofisticados, las aplicaciones móviles siguen siendo funcionales. Así es como los propietarios de aplicaciones móviles permiten a los consumidores seleccionar exactamente las funciones que deben tener sus dispositivos.

Existen diferentes tipos de aplicaciones móviles que responden a las necesidades y preferencias de los usuarios, así como a las capacidades técnicas de los dispositivos. A continuación, se detallan cada uno:

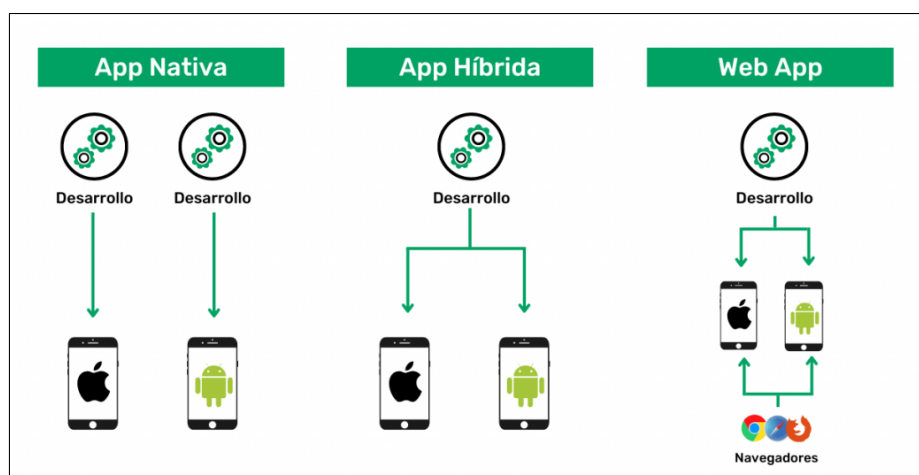


Figura 2.8: Tipo de aplicaciones móviles.

### 2.5.1. Aplicaciones nativas

Las aplicaciones nativas son apps desarrolladas para un sistema operativo móvil concreto (iOS o Android normalmente), en el lenguaje de programación específico de cada plataforma. Esto quiere decir que una app nativa creada para Android no puede ser utilizada en un dispositivo iOS y viceversa.

Es el tipo de aplicación móvil más conocida. Para que funcione, debemos descargarla desde los markets de apps, como App Store o Google Play e instalarla en nuestro teléfono [?].



## Ventajas

- **Tienen el mejor rendimiento.** Las aplicaciones nativas son las más rápidas y tienen un rendimiento superior a otros tipos de apps, ya que han sido optimizadas específicamente para el hardware y el sistema operativo del dispositivo.
- **Acceso completo e integración con las funciones hardware del dispositivo.** Las apps nativas permiten aprovechar al máximo las funcionalidades móviles: cámara, micrófono, lector biométrico de huella, sensores y redes inalámbricas.
- **Pueden funcionar sin acceso a internet (funcionamiento offline)** si han sido diseñadas para ello.

## Desventajas

- **Costes de desarrollo altos.** Si queremos tener nuestra app disponible para los dos sistemas, necesitaremos dos líneas de desarrollo diferentes, ya que el código utilizado para un sistema no es reutilizable para otro.
- **Complejidad de desarrollo.** Necesitamos equipos expertos en el lenguaje específico de cada sistema. Por ejemplo, en Kotlin para Android y en Swift para iOS.
- **Tiempo de desarrollo superior.** El desarrollo puede tomar entre 4 a 6 meses.

### 2.5.2. Aplicaciones Web

Las aplicaciones web realmente son webs especiales diseñadas para navegadores móviles. A diferencia de las apps nativas o híbridas, no necesitan ser descargadas, ya que se accede a ellas desde un navegador web.

Emplean las mismas tecnologías de desarrollo que una web, como HTML, CSS o JavaScript. Así, estaríamos hablando de una web con apariencia de app, por lo que presentaría sus mismas limitaciones. Sin embargo, con la llegada del HTML5, se han conseguido salvar algunas limitaciones, como el acceso a algunas funciones del móvil (geolocalización, cámaras).

## Ventajas

- **Carácter multiplataforma.** Con una sola línea de desarrollo.
- **Fácil desarrollo.** Se emplean tecnologías ampliamente conocidas.
- **Tiempo y coste de desarrollo bajo.**

## Desventajas

- **Acceso limitado a las funciones del dispositivo.**
- **No se pueden subir a las tiendas de aplicaciones.**
- **Diferentes experiencias de usuario.** Estas dependen del navegador utilizado.
- **Necesidad de conexión a Internet.** Incluso si se cuenta con un modo pensado para ello, es necesario para acceder a las posibles actualizaciones o para entrar por primera vez.

### 2.5.3. Aplicaciones Híbridas

Las aplicaciones híbridas o multiplataforma combinan elementos de las aplicaciones nativas y las aplicaciones web. Estas aplicaciones se desarrollan utilizando tecnologías web como HTML, CSS y JavaScript, pero se empaquetan en un formato que puede ser instalado en un dispositivo móvil como cualquier otra aplicación nativa. Por tanto, podemos obtener una aplicación para varias plataformas con un único desarrollo.

#### Ventajas

- **Menor coste.** Gracias al uso de lenguajes de programación más conocidos, con una mayor disponibilidad de profesionales en el mercado.
- **Carácter multiplataforma.** Con una sola línea de desarrollo.
- **Acceso a algunas funcionalidades del móvil.**
- **Reducción de los tiempos de desarrollo.** Generalmente, el tiempo de desarrollo se reduce a 3 meses.
- **Disponibilidad en markets.** Se pueden subir a los markets de aplicaciones, como App Store y Google Play.

#### Desventajas

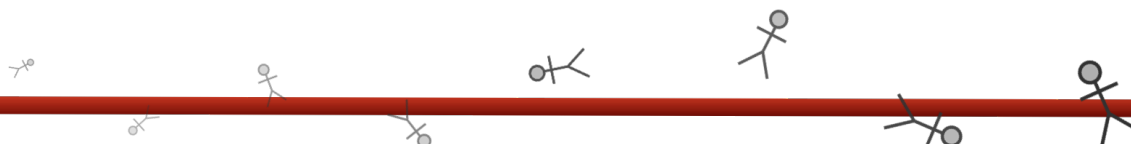
- **Rendimiento inferior.** Su rendimiento es inferior al de una app nativa, suelen tener un tamaño considerable y, además, ser más lentas.
- **Acceso limitado a las funciones del dispositivo.**

### 2.5.4. Aplicaciones Progresivas Web Apps (PWA)

Las aplicaciones progresivas son un reciente avance de las Web Apps. Al igual que las Web Apps, son webs diseñadas para móviles, pero esta vez, sí pueden ser descargadas en el móvil como una aplicación más, aunque no es necesario para que ofrezcan un comportamiento similar al de una app nativa a través del navegador.

Las PWA adoptan un comportamiento más propio de aplicaciones nativas que de web, como el funcionamiento sin Internet, un mayor rendimiento o su funcionamiento en segundo plano. Sin embargo, como desventaja, seguimos contando con la imposibilidad de subirlas a los markets de aplicaciones.

Para comprender mejor las diferencias entre los tipos de aplicaciones móviles, a continuación se presenta una tabla comparativa que destaca sus características clave:



Tipos de app	Nativa	Híbrida	Web
Interfaz	Basada en web	Específica de la plataforma (iOS, Android)	Basada en web
Tiempo de desarrollo	Alto	Medio	Bajo
Coste de desarrollo	Alto	Medio	Bajo
Multiplataforma	No	Sí	Sí
Rendimiento	Alto	Medio	Bajo
Acceso a los sensores del dispositivo	Completo	Alto o Completo	Limitado
Tiendas de aplicaciones	Sí	Sí	No

Cuadro 2.1: Comparación tipos de aplicaciones móviles.

Para el desarrollo de nuestro trabajo terminal, hemos decidido optar por una aplicación híbrida con el uso de Kotlin. Esta decisión se basa en varios factores relacionados con los recursos disponibles, las características de nuestro público objetivo y los plazos establecidos.

La aplicación móvil está dirigida para estudiantes, alumnos y personal de seguridad de la Escuela Superior de Cómputo, donde la mayoría utiliza dispositivos con sistema operativo Android. La elección de una aplicación híbrida nos permite optimizar la experiencia en Android, que es la plataforma que predomina entre nuestros usuarios.

Aunque las aplicaciones híbridas suelen desarrollarse con tecnologías web (como React Native o Flutter), para nuestro proyecto hemos decidido incorporar Kotlin para desarrollar modelos donde se requiera un rendimiento nativo o un acceso más profundo a las funciones del sistema operativo Android. Además, las aplicaciones híbridas permiten un desarrollo más rápido en comparación con las aplicaciones completamente nativas, ya que gran parte del código puede compartirse entre plataformas, así mismo, es una buena opción económica que se adapta a nuestro presupuesto.

## 2.6. Sistema operativo

## 2.7. Lenguaje de programación

## 2.8. Framework

### 2.8.1. Jetpack compose

Para la implementación de la aplicación móvil que forma parte de nuestro sistema de identificación y control de acceso, hemos decidido utilizar Jetpack Compose. La elección de la tecnología adecuada es importante para ofrecer una mejor experiencia de usuario y cumplir nuestros objetivos de diseño y funcional.

### ¿Por que Jetpack compose?

Jetpack compose es un framework (estructura o marco de trabajo que, bajo parámetros estandarizados, ejecutan tareas específicas en el desarrollo de un software) con la particularidad de ejecutar prácticas modernas en los desarrolladores de software a partir de la reutilización de componentes, así como también contando con la oportunidad de crear animaciones y temas oscuros. En este sentido, Jetpack Compose es el conjunto de herramientas ofrecidas por Android para el desarrollo de aplicaciones con un objetivo específico: simplificar y optimizar los códigos en la IU nativas [?].

## Ventajas

- **Menos código:** Simplifica el proceso de desarrollo haciendo menos código, todo se basa en funciones de modo que el código será simple y fácil de mantener.
- **Intuitiva:** Tan solo describe tu IU con un enfoque declarativo haciendo “qué hay que hacer” en vez de “cómo se debe hacer”.
- **Potente:** Tiene integrado Material Design con el cual puede crear apps atractivas al usuario con animaciones y mucho más.
- **Acelera el desarrollo:** Es compatible con proyectos existentes, puedes empezar a integrarlo por partes cuando quieras y donde quieras.
- **Kotlin:** Está escrito 100 % en Kotlin, lo cual nos permitirá usar sus herramientas potentes y API's intuitivas.

### 2.8.2. XML

Para este proyecto utilizaremos XML debido a la múltiples ventajas que ofrece en el desarrollo de interfaces de usuario en aplicaciones web y Android.

#### ¿Por que XML?

XML son siglas de *Extensible Markup Language*, es un lenguaje de marcado que proporciona reglas para definir cualquier dato.

Por ejemplo, imaginemos un documento de texto con comentarios. Los comentarios pueden ofrecer sugerencias como las siguientes:

- Ponga el título en negrita.
- Esta oración es un encabezado.
- Esta palabra es del autor.

Estos comentarios mejoran la usabilidad del documento sin repercutir en su contenido. Del mismo modo, XML utiliza símbolos de marcado para proporcionar más información sobre los datos.

#### Etiquetas XML

Los símbolos de marcado, denominados **etiquetas** en XML, se utilizan para definir los datos. Por ejemplo, para representar los datos de una librería, se pueden crear etiquetas como:

`<libro>`, `<título>` y `<autor>`

El documento XML de un solo libro tendría el siguiente contenido:

```
<libro>
<titulo>Introduccion a Amazon Web Services </titulo>
<autor>Mark Wilkins </autor>
</libro>
```

Las etiquetas ofrecen una sofisticada codificación de datos para integrar los flujos de información en diferentes sistemas.

## Ventajas de XML

- **Flexibilidad:** El formato XML es un lenguaje de marcas que se puede personalizar para diferentes propósitos.
- **Interoperabilidad:** El formato XML es compatible con una amplia gama de sistemas y aplicaciones, lo que significa que los datos se pueden intercambiar fácilmente entre diferentes sistemas.
- **Legibilidad:** El formato XML es fácil de leer y entender, lo que facilita la creación y el mantenimiento de archivos XML.
- **Reutilización:** Los elementos y atributos de un archivo XML se pueden reutilizar en diferentes partes del archivo, lo que ahorra tiempo y reduce errores.